DRAFT

Software Engineering Plan for the Land Information System

Submitted under Task Agreement GSFC-CT-2

Cooperative Agreement Notice (CAN) CAN-00OES-01

Increasing Interoperability and Performance of Grand Challenge Applications in the Earth, Space, Life, and Microgravity Sciences

June 14, 2002

DRAFT

Table of Contents

1. Introduction

This document describes the Software Engineering Plan for the Land Information System (LIS) to be implemented under funding from NASA's ESTO Computational Technologies (formerly High Performance Computing and Communications) Project. This plan is submitted to satisfy the Task Agreement GSFC-CT-2 under Cooperative Agreement Notice CAN-00-OES-01 Increasing Interoperability and Performance of Grand Challenge Applications in the Earth, Space, Life, and Microgravity Sciences.

1.1 Purpose

The Land Information System (LIS) project is to develop a leading-edge land surface modeling and data assimilation system, to support broad land surface research and application activities, to help define Earth system modeling interoperability standards, and to lead the effective application of Internet technology and cost-effective parallel processing to high-resolution, real-time Earth system studies.

1.2 Background

The Land Information System (LIS) is to be built upon the existing Land Data Assimilation Systems (LDAS), a pilot research effort of land surface modeling and data assimilation. LDAS has demonstrated successful land modeling capability and usability. However, LDAS can only operate with moderate spatial resolutions and no parallel processing support. This makes it impractical for potential high-resolution, real-time applications. LIS will inherit LDAS' land modeling and data assimilation capability, while adding parallel processing support, a web-based user interface, mass data storage, interoperability and portability, to push the envelope of the current land modeling effort towards to the goal of making a high-resolution, real-time, operation-quality land information system. The Land Information System will to enhance land surface research and applications and help define land surface modeling and assimilation standards. LIS will also assist in the definition and demonstration of the Earth System Modeling Framework (ESMF).

1.3 Organization and Responsibilities

The Hydrological Sciences Branch at NASA's Goddard Space Flight Center (GSFC) manages the LIS. The LIS development personnel are organized as two teams: the development team and the supporting team.

1.3.1 Development Team

The development team is responsible for the overall project management, team coordination, and development, maintenance, port and support of the core LIS system. The development team mainly consists of the team members at NASA/GSFC, with the following responsibilities:

- Dr. P. Houser: scientific supervision

- Dr. C. Peters-Lidard: project management, milestone enforcement, coordination, etc.
- Ms. S. Olden: System documentation, software engineering quality control.
- Dr. S. Kumar and Mr. J. Geiger: LDAS analysis and documentation, parallelization of LIS on Origin and Linux Cluster.
- Dr. Y. Tian: LIS hardware and software analysis and design. Leads a team of assemblers and system administrators.
- Mr. L. Lighty: Web interface design and implementation. Webmaster.

Drs. Houser and Peters-Lidard are serving as co-PI's on the project. They have substantial expertise in land modeling and the application of HPCC technologies to land-atmosphere modeling. More details about their background and experience is given below.

P. R. Houser (PI; NASA/GSFC) Dr. Houser's area of expertise is local to global land surface-atmospheric observation and numerical simulation, development and application of hydrologic data assimilation methods, and multi-scale soil moisture investigations. Dr. Houser's primary responsibility will be the management and coordination of the LIS team.

C. D. Peters-Lidard (Co-PI; NASA/GSFC) Dr. Peters-Lidard focuses on measurement and modeling of terrestrial water and energy balances and fluxes due to land-atmosphere interactions over a range of temporal and spatial scales. Dr. Peters-Lidard has extensive experience with massively parallel, coupled land-surface atmospheric models, and will have a primary responsibility in this area. Further, she will facilitate many aspects of clustered high performance computing, atmospheric coupling, earth system model connections, and advanced software engineering. Dr. Peters-Lidard will direct 5 full time computer scientist and software engineers working on LIS.

1.3.2 Supporting Team

The supporting team will supervise scientific issues, interoperability, land model development and data management utilities.

- Dr. P. Dirmeyer: ALMA, interface/coupling standards
- Mr. B. Doty: GrADS and GrADS-DODS Server
- Dr. K. Mitchell: NOAH Land Surface Model
- Dr. E. Wood and G. Goteti: VIC Land Surface Model
- Dr. S. Denning: Advisor on future carbon modeling

P. Dirmeyer (co-I; COLA) Dr. Dirmeyer conducts research into the role of land surface in the climate system. This includes development and application of land-surface models, studies of the impact of the land on the predictability of climate, and the impacts of land use change on regional and global climate. Dr. Dirmeyer is chair of the GEWEX global soil wetness project and as such has developed extensive community infrastructure frameworks and tools, such as the ALMA land surface model standards. Dr. Dirmeyer will be responsible for LIS software standards, the Community Land Model framework, will provide input to atmosphere coupling, earth system models, and software engineering. Dr. Dirmeyer will also serve as a collaboration mechanism with GEWEX global soil Wetness Project, ISLSCP, and COLA.

B. Doty (Co-I;COLA) Brian Doty has over 20 years of in-depth experience in a wide range of computing technologies. He developed the Grid Analysis and Display System (GrADS) package that is used worldwide for climate data analysis and visualization. He will be primarily responsible for the application of the GrADS-DODS Server (GDS) software to the LIS, in particular its support of remote users who wish to access and analyze high resolution land surface model output and remote sensing data sets. He will also initiate the work toward utilizing parallel data analysis techniques from within GrADS to permit rapid access and computation on the large data sets envisioned in this project.

K. Mitchell (Co-I;NCEP) "K. Mitchell (Co-I; NCEP) Dr. Mitchell is primarily responsible for land modeling and it's coupling to all regional and global weather and climate prediction models, and their supporting data assimilation systems, at the National Center for Environmental Prediction." He is the major developer of the NOAH land model, its NWP connections, and is a leader in the LDAS community framework. As part of an operational center, Dr. Mitchell has extensive experience with efficient land model coupled code implementation on very large compute platforms. Dr. Mitchell will be responsible for providing the NOAH model and guidance on its execution in the LIS, including supporting documentation, ancillary files, example control runs, and cold-start initialization strategies. Additionally, Dr. Mitchell will advise on LIS atmosphere connections and provide feedback from the North American LDAS project, including validation and assimilation approaches, as well as software utilities."

E. Wood (Co-I;Princeton) Professor Wood’s specialty is hydroclimatology, with research focuses in land-atmospheric interactions, terrestrial remote sensing, hydrologic impacts from climate change, and environmental data analysis. Professor Wood will be responsible for the inclusion of the VIC model in LIS, various aspects of LIS remote sensing, and general hydrologic modeling and analysis. Professor Wood has one full time computer scientist working on the VIC model and LIS implementation of the model.

S. Denning (Co-I; Colorado State University) Professor Denning investigates the interface between terrestrial ecosystems and the atmosphere, with particular emphasis on using atmospheric observation to understand the global carbon cycle. His research involves the development and application of simulation models of these constraints on the carbon budget of the atmosphere. He emphasized the “fusion” of modeling and data analysis to understand natural processes. Professor Denning will be advise on the land surface biogeochemistry and carbon cycle remote sensing and modeling aspects of LIS.

1.3.3 Interfacing Groups

LIS will interface with a variety of groups in addition to the Global Land Data Assimilation System, the North American Land Data Assimilation System, the DODS/GrADS, and the Community Land Model community, the NOAH Land Model community and the VIC Land Model community. Other interfacing groups will include the Earth System Modeling Framework (ESMF) community, the NASA ESTO Computational Technologies Project, John Dorband's Linux cluster group at GSFC, and the ALMA standards group.

2. LIS Problem Statement

The Land Information System (LIS) will have the following components: (1) A high-resolution (1km) global Land Data Assimilation System (LDAS), involving several independent community Land Surface Schemes (LSS), land surface data assimilation technologies, and integrated database operations for observation and prediction data management; and (2) A web-based user interface based on the Grid Analysis and Display System (GrADS) and the Distributed Oceanographic Data System (DODS) for accessing data mining, numerical modeling, and visualization tools. The LIS will be available both as a portable system for download together with a small database to an independent system for use in smaller applications, and as a "production" system on a centralized server for large applications. By incorporating and promulgating the existing Assistance for Land surface Modeling Activities (ALMA; http://www.lmd.jussieu.fr/ALMA/) and DODS standards for model coupling and data visualization, LIS will contribute to the definition of the land surface modeling and assimilation standards for the Earth System Modeling Framework (ESMF).

LIS has to face the challenges in the following six areas,

- Modeling reality. This is a test of our scientific understanding of the physical processes of the global land surface. Fortunately, the LIS developer team has the leading researchers in the land surface modeling community, and the scientific issues have been attacked in the existing LDAS project extensively, so we are confident that the LIS will be built on a solid scientific foundation. Moreover, the development of LIS and its high-quality data will help us to advance our scientific research on land modeling.
- Performance. The requirements of near real-time, high-resolution demands high performance computing. From base-lining the LDAS runs, we have obtained realistic estimate of the required computing power for LIS, and have gained insight on efficient parallelization schemes we will use for LIS. We will both build a cost-effective Linux cluster, and use NASA's existing Origin supercomputers, with parallelized LIS code, to meet the demand.
- Data storage. The high-resolution LIS system will produce huge amount of data. To manage this issue, we will take advantage of our Linux cluster's distributed storage feature, to store output data in a clustered environment to ensure reliable, high-performance data management operations.
- Usability. LIS will be a sophisticated system. To let users access LIS, we will design intuitive web-based interfaces, and users can have different levels of access, depending on their needs. Meanwhile, we will extensively employ GrADS's graphical capability and GrADS/DODS server's data sub-setting functionality to enable users efficient access and use of the LIS data.
- Interoperability. LIS is required to be able to interface with other Earth system models. This can be achieved by incorporating the ALMA (Assistance for Land

surface Modeling Activities) standard, and by following and contributing to the ESMF (Earth System Modeling Framework) standard.
- Portability. LIS has to be able to run both on NASA's Origin supercomputers, and on our custom-built Linux cluster. This is easy to achieve, since the LIS source code for scientific computing will be written in the high-level languages, Fortran and C, with MPI libraries. Since Fortran, C and MPI are all highly standard and portable, as long as we refrain from using any platform-specific language features, we will deliver LIS as a portable system.

In summary, developing LIS presents quite a few challenges, and we are confident that our skills, experience and development strategy will be able to handle them to achieve our goal, and in the meantime, contribute to the knowledge and advancement in these areas.

3. Technical Approach

3.1 Assumptions and Constraints

This plan is based on, and constrained by the following factors:

- Our scientific understanding of the land surface physics and processes. We are under the assumption that based on the current state of the art science understanding of the macroscale water, energy and momentum transfer processes, we will be able to realistically reproduce the Earth's land surface and its dynamics in our computer models. However, the land surface processes are highly complicated, and our success in this aspect is limited by the current scientific knowledge in this area.
- The development, validation and operation of LIS will heavily rely on the availability and quality of the forcing and observation data. These data fundamentally dictate the LIS output data.
- Sufficient and dedicated computing resources. We expect the NASA/ARC Origin supercomputing facility will provide us with a suitable development environment and ample computing power. Also the LIS development and operation depends critically on the successful installation and smooth operation of our Linux cluster within the constraint of the budget and human resources.

3.2 Development Environment

The primary hardware platform will be a Linux cluster with approximately 200 PC nodes, running standard Linux kernel. A secondary hardware platform is SGI Origin 2000 (and eventually Origin 3000) running IRIX, used as a test bed for prototyping and verification of software portability.

Scientific computation of LIS on both platforms will be developed mainly in FORTRAN 90 and C. Parallel computing will be implemented with MPI. System management and job scheduling will be done with third-party open source Unix/Linux packages, supplemented with software utilities developed in-house. User interfaces will be largely web-based, developed in static HTML or interactive CGI scripts, with Cascade Style Sheets (CSS) for page layout.

3.3 Activities, tools, and products

The development activities will revolve around the standard software development cycle: requirement analysis, system design, developing, testing and maintenance:

- Requirement analysis. This will start from detailed performance analysis and baselining of the existing LDAS system. The performance and resource usage information gathered will enable us to predict the requirements for LIS. The product of the requirement analysis will be LDAS performance data and LDAS source code documentation, as well as LIS requirement projections resulted from this analysis. In addition, the implication of LIS' real-time, high-resolution requirements will also be analyzed and documented. The analysis will be submitted via web before project milestone (E).
- System design. We will employ a systematic, top-down approach for the design of the system. We will start with the top level system physical and logical architecture, working our way down the various system components and sub-components. Industry standard design tools, such as Microsoft Visio, will be used. The system design will produce detailed diagrams and specifications of the system, including hardware layout and connection, software structure and flow chart, network design, and data management infrastructure. The result of the design will be detailed design documentation in non-proprietary format, and will be submitted to meet project milestone (E).
- Development. Standard coding practices and styles will be followed. The source code will be extensively documented both in-line and stand-alone. The source code will be accompanied by detailed documentation, including industry-standard diagrams such as calling-tree diagrams and data flow diagrams. All the development and code improvement will be delivered at the corresponding LIS milestones.
- Testing. We will carry out the following testing activities at every proper stage of the development cycle:

    a. Component testing. Each component will be tested to ensure validity and optimal performance.

    b. Integration testing. We will make sure all the pieces can work together, produce correct results and achieve maximum performance.

    c. Validation testing. The system output will be tested against known output from LDAS and compare with observational data.

    d. Job stream testing. At every stage of the development, we will make sure the system will work smoothly in a production environment, from data input, computation, data analysis, data output all the way to user interaction.

    e. Interface testing. Tests will be done to make sure LIS will interface with the data suppliers, the users and land surface models consistently.

    f. Security testing. The system will be tested against real-word situations where incorrect input, user errors, malicious users, software bugs, to correct any potential security hazards.

    g. Stability and recovery testing. The system will be tested to verify great fault tolerance and graceful recovery.

h. Performance testing. Performance will be measured on both platforms to see if the performance requirements are met, and provide feedback for bottleneck identification and fine tuning.

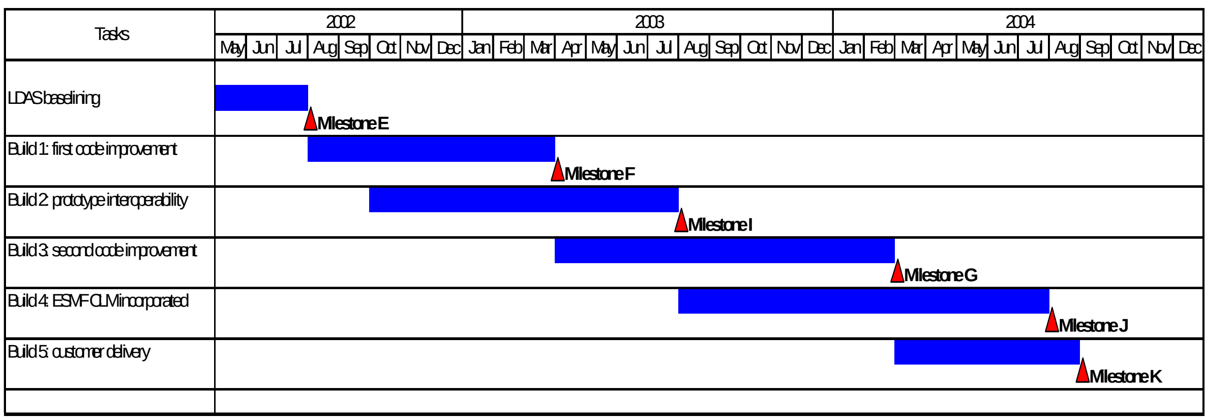
The initial version of the test plan will be delivered via web at LIS milestone (H). A detailed testing report will be produced at every stage of the development cycle to guide further development.

- Maintenance. The source code will be maintained by the CVS system. Regular software and data backup will be employed.

3.4 Build Strategy

The LIS build strategy follows the milestones set forth in for the project. Table 1 below illustrates the five code delivery milestones and associated builds, including the baselining of the original code. Each build is described in more detail below.

Table 1. LIS milestones and build strategy.



The initial cycle of the software development is to move the GLDAS, NOAH, and CLM to the ESS Tested. At this stage, there will be no design changes. The expected changes will be to make the software compatible with the existing ESS Testbed operating system and operating environment. This Global LDAS code baseline will be completed by July 30, 2002.

Build 1 will be the first code improvement. This is in conjunction with Milestone F that will be completed by March 30, 2003. The expected changes will be to parallelize a portion of the code to generate better performance. Build 1 will be on the ESS Tested.

Build 2 will consist of generating a scheme for interoperability and community requirements and demonstrate a prototype. This is milestone I) which will be completed by July 30, 2003. Build 2 will demonstrate the VIC code on both the ESS Tested and the LIS cluster.

Build 3 will make the second code improvement, which will add the database/data management functions and the visualization function. This will be the implementation of the GRADS/DODS servers with the Internet based user interface. The code will also be optimized for faster processing time. All of these functions take place on the LIS cluster. This corresponds with milestone G that will be completed by Feb 30, 2003.

Build 4 is based on receiving an ESMF compliant version of the CLM code by June 2003. For milestone J) LIS would need to implement the CLM code into the LIS. This task will be completed by July 30, 2004.

Build 5 is the last version of the LIS under milestone K that is finished in August of 2004. For the final product, the LIS will need to run the ESMF version of CLM and the existing LIS/local versions of CLM, NOAH, and VIC.

4. Management Approach

4.1 Assumptions and Constraints

The Hydrological Sciences Branch at NASA's Goddard Space Flight Center will manage the LIS project. The software engineering and development will be managed by Dr. Christa Peters-Lidard. Each partner in the project will work from his or her home organizations' facility. Weekly status meetings will be held where any of the collaborating partners can teleconference in. Monthly status teleconferences will be held where a representative of each of the collaborating partners are encouraged to attend. Other working group meetings will be scheduled as needed.

The Land Surface Models CLM, VIC, and NOAH will not be managed by LIS. These communities have established procedures for successfully managing publicly available software. We will interface with each of these groups by having a member of the LIS team as an established member of these communities and by scheduling working group meetings as needed.

The DODS/GrADS team from COLA also has established procedures for managing publicly available software. COLA will be developing additional capabilities for LIS that will subsequently be available to the wider DODS/GrADS community. We will interface directly with P. Dirmeyer and B. Doty of COLA to coordinate COLA's effort.

4.2 Milestones and Schedules

These are the milestones and the not later than dates for the LIS:

Project Start	01-FEB-2002
A) Software Engineering Plan	30-JUN-2002
E) Global LDAS Code Baselined	30-JUL-2002
B) FY02 Annual Report delivered	30-AUG-2002
f) Installation of Linux Cluster	30-AUG-2002
H) Design policy for Interoperability and Community Delivery	28-FEB-2003

F) First Code Improvement	30-MAR-2003
I) Interoperability Prototype	30-JUL-2003
C) FY03 Annual Report	30-AUG-2003
G) Second Code Improvement	28-FEB-2003
J) Full Interoperability demonstrated	30-JUL-2004
K) Customer delivery Accomplished	30-AUG-2004
K) Present LIS to Review Board	30-AUG-2004
D) Final Report Delivered	28-FEB-2005

4.3 Metrics

LIS will generate metrics on the software residing on the ESS Testbed, the software on the LINUX cluster, and useage metrics on the LINUX cluster when it becomes available. Various metrics that deal with the measurement of the software product will be employed. The metrics planned for the estimating the size of the product are Total lines of code broken down into lines of executable code, comment lines, and blank lines, and number of modules (subroutines). Metrics such as cyclomatic complexity and level of nesting will be used as a measure of complexity of the code. These metrics will be reported for both the Linux cluster and the ESS Testbed. We will also report on the LIS System usage of the ESS Testbed monthly.

LIS will operate a web-based defect tracking tool. The main point of the tool will be to track any known defect to closure. We plan to have a simple metric on number of defects per month, number of defects closed, and average time to close defects.

After the LIS Cluster is operational, the metrics planned are the typical metrics used for system administration and management such as system availability, processor time used, processor time idle, number of users, etc. and other metrics to allow LIS to tune the system to be most efficient.

4.4 Risk Management

LIS plans to practice risk management to anticipate, mitigate and control high impact risks to the project. A high impact risk is a technical risk causing loss of the project, loss of critical function, or science objective or a schedule slip of 6 months or more beyond the scheduled milestone. High impact risks will be continuously brought to the attention of the Principal Investigator. All high impact risks will be assigned to an engineer to generate a mitigation plan and track the action to closure. The mitigation plans will be approved, modified, or rejected at the next scheduled weekly status meeting and status of the mitigation plan will be an agenda item for the monthly teleconference. Any open high impact risks will be tracked until the risk is resolved or no longer considered high impact. The risks to the LIS project will be reviewed at the beginning of each new milestone period.

5. Product Assurance

5.1 Configuration Management

The LIS software will be managed using the tool Concurrent Versions System (CVS). CVS provides a repository for code, checkout of code, a history of changes made to the code, and a commit feature to prevent developers from making contradictory changes to the same code.

LIS will have an assigned gatekeeper and backup gatekeeper to accept changes to the baseline version of software. All software will have version numbers. When using Land Surface Models such as CLM and NOAH that are in the public domain, LIS will continue using the version number that was assigned by the developing organization.

Additional documentation on CVS can be obtained from http://www.gnu.org/manual/cvs-1.9/cvs.html.

LIS will have a documentation repository on the public web site. The released version of each document will be displayed on the site. Non-current versions of the documentation will be stored in a simple file directory on the server. The Webmaster will control access to the web site.

5.2 Quality Assurance

The LIS will address quality in all aspects of the program. We will concentrate on conducting formal technical reviews and performing well-planned software testing. Our goal will be to prevent defects if possible, and detect and remove remaining defects as early in the life cycle of the software as possible.

The technical review most advantageous to the LIS project will be code walkthroughs. We will walkthrough all new code generated as part of this project. The preferred method is to walkthrough small portion of the code with a small group of peers.

LIS will also conduct a Requirements review, a Software Design Review, an Interoperability Design Review, and Interface (GUI) design review, and a Test Plan Review. Each of these reviews will be held with the LIS Principal Investigators. Each review will result in review minutes that will include a list of attendees, a list of action items, and the person that is assigned to respond to the action item. The review minutes will be stored on the LIS intranet area.